CLAIMS

1. A block interleaving apparatus comprising:

a storage means to which (L×M) pieces of addresses are allocated (L,M: integers, 2≤L,M);

an address generation means for generating addresses for writing and reading blocks, each block having (L×M) pieces of data as a unit to be subjected to block interleaving, in/from the storage means; and

a control means for controlling the storage means so that the storage means switches the operation between the data writing and the data reading, by using the addresses generated by the address generation means;

said address generation means comprising:

a multiplication means for generating the product of α (α: integer, 2≤) and $M^{(b-x)}$ (x: integer, 0≤x≤b, b: integer, 0≤b), every time a block of a block number b is inputted;

a first overflow processing means having a first comparison means for comparing the product obtained by the multiplication means with a comparison reference value L×M-1, and subtracting, as much as possible, the L×M-1 from the product on the basis of the result of the comparison to suppress overflow of the product, thereby outputting an address increment value REG corresponding to of the block having the block number b;

an addition means for successively adding the (n-1)th

(n: integer, $1 \leq n \leq L \times M-1$) address Ab(n-1) of the block having the block number b, to the address increment value REG outputted from the first overflow processing means, every time the block of the block number b is inputted, thereby successively generating the n-th address Ab(n) in the block of the block number b; and

a second overflow processing means having a second comparison means for comparing the sum obtained by the addition means with the comparison reference value $L \times M-1$, and subtracting, as much as possible, the $L \times M-1$ from the sum on the basis of the result of the comparison to suppress overflow of the sum, thereby outputting an address to be actually supplied to the storage means;

wherein, when the first comparison means compares the product obtained by the multiplication with the comparison reference value $L \times M-1$, the first comparison means employs, as a comparison reference value instead of the $L \times M-1$, the minimum value A which exceeds the $L \times M-1$ and is included in the product.

2. A block interleaving apparatus comprising:

a storage means to which $(L \times M)$ pieces of addresses are allocated (L,M: integers, $2 \leq L,M$);

an address generation means for generating addresses for writing and reading blocks, each block having $(L \times M)$ pieces of data as a unit to be subjected to block interleaving, in/from the storage means; and

a control means for controlling the storage means so that the storage means switches the operation between the data writing and the data reading, by using the addresses generated by the address generation means;

said address generation means including:

an address increment value storage means for storing an address increment value REG(b) corresponding to a block having a block number b (b: integer, 1≤b);

a first initial value setting means for setting α (α: integer, 2≤) as an address increment value REG(0) corresponding to a block having a block number 0, in the address increment value storage means;

a multiplication means for multiplying the output value REG(c) (c=b-1) from the address increment value storage means by M;

a first overflow processing means having a first comparison means for comparing the product obtained by the multiplication means with a comparison reference value L×M-1, and subtracting, as much as possible, the L×M-1 from the product on the basis of the comparison result to perform a calculation equivalent to "α×M**(b-x)mod(L×M-1)" (M**(b-x) means $M^{(b-x)}$, mod is the remainder, x is an integer, 0≤x≤b), thereby suppressing overflow, and outputting the calculation result as an address increment value REG(b) corresponding to the block of the block number b to the address increment value storage means;

an address storage means for storing the n-th (n: integer, $1 \leq n \leq L \times M-1$) address Ab(n) in the block of the block number b (b: integer, $1 \leq b$), and outputting it to an address input terminal of the storage means;

a second initial value setting means for setting the 0th address Ab(0) corresponding to the block of the block number b in the address storage means;

an addition means for adding the address increment value REG(b) from the address increment value storage means, to the output value Ab(p) (p=n-1) from the address storage means;

a second overflow processing means having a second comparison means for comparing the sum obtained by the addition means with the comparison reference value $L \times M-1$, and subtracting, as much as possible, the $L \times M-1$ from the sum on the basis of the comparison result to perform a calculation equivalent to "$(Ab(n-1)+\alpha \times M^{**}(b-x)) \bmod (L \times M-1)$", thereby suppressing overflow of the sum, and outputting the calculation result as the n-th address Ab(n) of the block having the block number b to the address storage means;

wherein, when the first comparison means compares the product obtained by the multiplication with the comparison reference value $L \times M-1$, the first comparison means employs, as a comparison reference value instead of the $L \times M-1$, the minimum value A which exceeds the $L \times M-1$ and is included in the product.

3. The block interleaving apparatus of Claim 2 wherein,

said first initial value setting means comprises:

a first constant generation means for generating the α; and

a first selector for selecting the α from the first constant generation means when a reset signal is inputted, and outputting it to the address increment value storage means;

said first overflow processing means comprises:

a second selector for receiving the output of the multiplication means and the output of the address increment value storage means, and selecting the output of the multiplication means at the beginning of each block, and selecting the output of the address increment value storage means during a period of time other than the beginning of the block;

a first comparison means for comparing the output of the second selector with the comparison reference value A;

first subtraction means for subtracting the $L \times M-1$ from the output of the second selector; and

a third selector for receiving the output of the second selector and the output of the first subtraction means, and selecting the output of the first subtraction means when the output of the second selector is equal to or larger than the comparison reference value, and selecting the output of the second selector when the output of the second selector is smaller than the comparison reference value;

wherein the output of the third selector is supplied to the address increment value storage means through the first selector during a period of time when the reset signal is not inputted.

4. The block interleaving apparatus of Claim 2 wherein said first comparison means employs, as a comparison reference value instead of the minimum value A exceeding the L×M-1, a value B which satisfies L×M-1<B<A and it selected so that the number of logic gates constituting the comparison means is minimized.

5. The block interleaving apparatus of Claim 2 wherein,

said second initial value setting means comprises:

a second constant generation means for generating a value 0; and

a fourth selector for selecting the value 0 from the second constant generation means when a reset signal is inputted, and outputting it to the address storage means;

said second overflow processing means comprises:

a second comparison means for comparing the output of the addition means with the comparison reference value L×M-1;

a second subtraction means for subtracting the comparison reference value L×M-1 from the output of the addition means; and

a fifth selector for receiving the output of the

addition means and the output of the second subtraction means, and selecting the output of the second subtraction means when the output of the addition means is equal to or larger than the comparison reference value, and selecting the output of the addition means when the output of the addition means is smaller than the comparison reference value;

wherein the output of the fifth selector is supplied to the address storage means through the fourth selector during a period of time when the reset signal is not inputted.

6. The block interleaving apparatus of Claim 2 wherein the values of α and $L \times M-1$ are set so that no common divisor exists between them.

7. The block interleaving apparatus of Claim 2 wherein the values of α and $M^{(-x)}$ are set so that α is not equal to $M^{(-x)}$.

8. The block interleaving apparatus of Claim 2 wherein the values of α, L, and M are set at 20, 8, and 203, respectively.

9. The block interleaving apparatus of Claim 2 wherein the values of (L,M) are set at any of 72 possible values as follows:

$L=96 \times X$ $(X=1,2,4)$, $M=2,\ldots,13$, or

$M=2,\ldots,13$, $L=96 \times X$ $(X=1,2,4)$.

10. A block deinterleaving apparatus comprising:

a storage means to which (L×M) pieces of addresses are allocated (L,M: integers, 2≤L,M);

an address generation means for generating addresses for writing and reading blocks, each block having (L×M) pieces of data as a unit to be subjected to block interleaving, in/from the storage means; and

a control means for controlling the storage means so that the storage means switches the operation between the data writing and the data reading, by using the addresses generated by the address generation means;

said address generation means comprising:

a multiplication means for generating the product of α (α: integer, 2≤) and $L^{(b-x)}$ (x: integer, 0≤x≤b, b: integer, 0≤b), every time a block of a block number b is inputted;

a first overflow processing means having a first comparison means for comparing the product obtained by the multiplication means with a comparison reference value L×M-1, and subtracting, as much as possible, the L×M-1 from the product on the basis of the comparison result to suppress overflow of the product, thereby outputting an address increment value REG corresponding to the block having the block number b;

an addition means for successively adding the (n-1)th (n: integer, 1≤n≤L×M-1) address Ab(n-1) of the block having the block number b, to the address increment value REG outputted from

the first overflow processing means, every time the block of the block number b is inputted, thereby successively generating the n-th address Ab(n) in the block of the block number b; and

a second overflow processing means having a second comparison means for comparing the sum obtained by the addition means with the comparison reference value L×M-1, and subtracting, as much as possible, the L×M-1 from the sum on the basis of the comparison result to suppress overflow of the sum, thereby outputting an address to be actually supplied to the storage means;

wherein, when the first comparison means compares the product obtained by the multiplication with the comparison reference value L×M-1, the first comparison means employs, as a comparison reference value instead of the L×M-1, the minimum value A which exceeds the L×M-1 and is included in the product.

11. A block deinterleaving apparatus comprising:

a storage means to which (L×M) pieces of addresses are allocated (L,M: integers, 2≤L,M);

an address generation means for generating addresses for writing and reading blocks, each block having (L×M) pieces of data as a unit to be subjected to block interleaving, in/from the storage means; and

a control means for controlling the storage means so that the storage means switches the operation between the data writing and

the data reading, by using the addresses generated by the address generation means;

said address generation means including:

an address increment value storage means for storing an address increment value REG(b) corresponding to a block having a block number b (b: integer, 1≤b);

a first initial value setting means for setting α (α: integer, 2≤) as an address increment value REG(0) corresponding to a block having a block number 0, in the address increment value storage means;

a multiplication means for multiplying the output value REG(c) (c=b-1) from the address increment value storage means by L;

a first overflow processing means having a first comparison means for comparing the product obtained by the multiplication means with a comparison reference value L×M-1, and subtracting, as much as possible, the L×M-1 from the product on the basis of the comparison result to perform a calculation equivalent to "α×L**(b-x)mod(L×M-1)" (L**(b-x) indicates $L^{(b-x)}$, mod is the remainder, x is an integer, 0≤x≤b), thereby suppressing overflow, and outputting the calculation result as an address increment value REG(b) corresponding to the block of the block number b to the address increment value storage means;

an address storage means for storing the n-th (n: integer, 1≤n≤L×M-1) address Ab(n) in the block of the block

number b, and outputting it to an address input terminal of the storage means;

a second initial value setting means for setting the 0th address Ab(0) of the block of the block number b in the address storage means;

an addition means for adding the address increment value REG(b) from the address increment value storage means to the output value Ab(p) (p=n-1) from the address storage means;

a second overflow processing means having a second comparison means for comparing the sum obtained by the addition means with the comparison reference value L×M-1, and subtracting, as much as possible, the L×M-1 from the sum on the basis of the comparison result to perform a calculation equivalent to "(Ab(n-1)+α×L**(b-x))mod(L×M-1)", thereby suppressing overflow of the sum, and outputting the calculation result as the n-th address Ab(n) corresponding to the block having the block number b to the address storage means;

wherein, when the first comparison means compares the product from the multiplication means with the comparison reference value L×M-1, the first comparison means employs, as a comparison reference value instead of the L×M-1, the minimum value A which exceeds the L×M-1 and is included in the product.

12. The block deinterleaving apparatus of Claim 11 wherein, said first initial value setting means comprises:

a first constant generation means for generating the α; and

a first selector for selecting the α from the first constant generation means when a reset signal is inputted, and outputting it to the address increment value storage means;

said first overflow processing means comprises:

a second selector for receiving the output of the multiplication means and the output of the address increment value storage means, and selecting the output of the multiplication means at the beginning of each block, and selecting the output of the address increment value storage means during a period of time other than the beginning of the block;

a first comparison means for comparing the output of the second selector with the comparison reference value A;

a first subtraction means for subtracting the $L \times M-1$ from the output of the second selector; and

a third selector for receiving the output of the second selector and the output of the first subtraction means, and selecting the output of the first subtraction means when the output of the second selector is equal to or larger than the comparison reference value, and selecting the output of the second selector when the output of the second selector is smaller than the comparison reference value;

wherein the output of the third selector is supplied to the address increment value storage means through the first

selector during a period of time when the reset signal is not inputted.

13. The block deinterleaving apparatus of Claim 11 wherein said first comparison means employs, as a comparison reference value instead of the minimum value A exceeding the L×M-1, a value B which satisfies L×M-1<B<A and is selected so that the number of logic gates constituting the comparison means is minimized.

14. The block deinterleaving apparatus of Claim 11 wherein,

said second initial value setting means comprises:

a second constant generation means for generating a value 0; and

a fourth selector for selecting the value 0 from the second constant generation means when a reset signal is inputted, and outputting it to the address storage means;

said second overflow processing means comprises:

a second comparison means for comparing the output of the addition means with the comparison reference value L×M-1;

a second subtraction means for subtracting the comparison reference value L×M-1 from the output of the addition means; and

a fifth selector for receiving the output of the addition means and the output of the second subtraction means, and selecting the output of the second subtraction means when the

output of the addition means is equal to or larger than the comparison reference value, and selecting the output of the addition means when the output of the addition means is smaller than the comparison reference value;

wherein the output of the fifth selector is supplied to the address storage means through the fourth selector during a period of time when the reset signal is not inputted.

15. The block deinterleaving apparatus of Claim 11 wherein the values of α and $L \times M-1$ are set so that no common divisor exists between them.

16. The block deinterleaving apparatus of Claim 11 wherein the values of α and $L^{(-x)}$ are set so that α is not equal to $L^{(-x)}$.

17. The block deinterleaving apparatus of Claim 11 wherein the values of α, L, and M are set at 20, 8, and 203, respectively.

18. The block deinterleaving apparatus of Claim 11 wherein the values of (L,M) are set at any of 72 possible values as follows:

$L=96 \times X$ $(X=1,2,4)$, $M=2,\ldots,13$, or

$M=2,\ldots,13$, $L=96 \times X$ $(X=1,2,4)$.

19. A block interleaving method for performing block interleaving of data by generating addresses for writing and

reading blocks, each block having (L×M) pieces of data (L,M: integers, 2≤L,M) as a unit to be interleaved, in/from a storage means to which (L×M) pieces of addresses are allocated, and controlling the storage means by using the generated addresses so that the storage means switches the operation between the data writing and the data reading:

wherein, α (integer, 2≤) is given as an address increment value REG to a block having a block number 0 and, thereafter, the increment value REG is multiplied by M every time the block number increments by 1 and thus obtained REG is used as an address increment value REG of the corresponding block, and when the address increment value REG exceeds L×M-1, the remainder over L×M-1 is used as an increment value instead of the increment value REG to repeat the above-described processing, thereby performing a calculation equivalent to "α×M**(b-x)mod(L×M-1)" (M**(b-x) indicates $M^{(b-x)}$, mod is the remainder, and x is an integer, 0≤x≤b) to obtain an address increment value of each block;

in the case where Ab(0) is set as an initial value of address in each block and, thereafter, the address increment value REG in this block is successively summed to generate addresses Ab(1) to Ab(n) (n: integer, 1≤n≤L×M-1) in this block, when the address exceeds L×M-1, the remainder over L×M-1 is used as an address instead of the address to repeat the above-described processing, thereby generating addresses in each block;

and

when calculating the address increment value, decision as to whether the remainder is to be obtained or not is made by comparing the address increment value with the L×M-1 using first comparison means and, at this time, the minimum value A which exceeds the L×M-1 and is included in the result of multiplication is used as a comparison reference value instead of the L×M-1.

20. The block interleaving method of Claim 19 wherein said first comparison means employs, as a comparison reference value instead of the minimum value A exceeding the L×M-1, a value B which satisfies $L \times M-1 < B < A$ and is selected so that the number of logic gates constituting the comparison means is minimized.

21. The block interleaving method of Claim 19 wherein the values of α and L×M-1 are set so that no common divisor exists between them.

22. The block interleaving method of Claim 19 wherein the values of α and $M^{(-x)}$ are set so that α is not equal to $M^{(-x)}$.

23. The block interleaving method of Claim 19 wherein the values of α, L, and M are set at 20, 8, and 203, respectively.

24. The block interleaving method of Claim 19 wherein the values of (L,M) are set at any of 72 possible values as follows:

L=96×X (X=1,2,4), M=2,...,13, or

M=2,...,13, L=96×X (X=1,2,4).

25. A block deinterleaving method for performing block deinterleaving of data by generating addresses for writing and reading blocks, each block having (L×M) pieces of data (L,M: integers, 2≤L,M) as a unit to be deinterleaved, in/from storage means to which (L×M) pieces of addresses are allocated, and controlling the storage means by using the generated addresses so that the storage means switches the operation between writing and reading of the data:

wherein, α (integer, 2≤) is given as an address increment value REG to a block having a block number 0 and, thereafter, the increment value REG is multiplied by L every time the block number increments by 1 and thus obtained REG is used as an address increment value REG of the corresponding block, and when the address increment value REG exceeds L×M-1, the remainder over L×M-1 is used as an increment value instead of the increment value REG to repeat the above-described processing, thereby performing a calculation equivalent to "α×L**(b-x)mod(L×M-1)" (L**(b-x) indicates $L^{(b-x)}$, mod is the remainder, and x is an integer, 0≤x≤b) to obtain an address increment value of each block;

in the case where Ab(0) is set as an initial value of address in each block and, thereafter, the address increment value REG in this block is successively summed to generate addresses Ab(1) to Ab(n) (n: integer, $1 \leq n \leq L \times M-1$) in this block, when the address exceeds $L \times M-1$, the remainder over $L \times M-1$ is used as an address instead of the address to repeat the above-described processing, thereby generating addresses in each block; and

when calculating the address increment value, decision as to whether the remainder is to be obtained or not is made by comparing the address increment value with the $L \times M-1$ using first comparison means and, at this time, the minimum value A which exceeds the $L \times M-1$ and is included in the result of multiplication is used as a comparison reference value instead of the $L \times M-1$.

26. The block deinterleaving method of Claim 25 wherein said first comparison means employs, as a comparison reference value instead of the minimum value A exceeding the $L \times M-1$, a value B which satisfies $L \times M-1<B<A$ and is selected so that the number of logic gates constituting the comparison means is minimized.

27. The block deinterleaving method of Claim 25 wherein the values of α and $L \times M-1$ are set so that no common divisor exists between them.

28. The block deinterleaving method of Claim 25 wherein the values of α and $L^{(-x)}$ are set so that α is not equal to $L^{(-x)}$.

29. The block deinterleaving method of Claim 25 wherein the values of α, L, and M are set at 20, 8, and 203, respectively.

30. The block deinterleaving method of Claim 25 wherein the values of (L,M) are set at any of 72 possible values as follows:

$L=96\times X$ (X=1,2,4), M=2,...,13, or

M=2,...,13, $L=96\times X$ (X=1,2,4).